on the programmable controller. The programmable controller is hereafter referred to as interpreter-type programmable controller.

Amendments to the paragraph beginning at page 3, line 22:

Moreover, a compiling-type programmable controller is practically used as another type of programmable controller that receives directly-executable ~~codes~~ code from a control-program-development supporting apparatus ~~by~~ through a microprocessor ~~mounted on~~ in the programmable controller and executes the codes. In the case of the configuration using the compiling-type programmable controller, a control-program-development supporting apparatus is provided with a compiler, which converts a ladder diagram directly or temporarily into an instruction list by the compiler and then, compiles the list into code directly executable ~~codes~~ by the microprocessor and transfers the execution codes to the programmable controller.

Amendments to the paragraph beginning at page 4, line 20:

However, because the above control program depends on the ~~bender~~ vendor or type of ~~a~~ programmable controller, it is impossible to execute the same sequential processing between different types of programmable controllers by using control programs having the same source code. Therefore, it is necessary to develop a new control program or transplant a new model or different model to a programmable controller and thus, it is difficult to effectively use past properties of a control program.

Amendments to the paragraph beginning at page 5, line 1:

For example, according to the advanced-language sequence instruction program generator and universal sequencer disclosed in Japanese Patent Laid-Open No. HEI 7-295612, the advanced-language sequence instruction program generator translates a ~~sequence instruction~~ language sequence instruction program into an advanced-language sequence instruction program with a translation program and compiles the advanced-language sequence instruction program to generate an execution-type sequence instruction program. Or, the universal sequencer executes the advanced-language sequence instruction program while sequentially interpreting the program. Thereby, it is

possible to execute sequential control ~~by~~ with the same control program independently of the ~~bender~~ vendor or type of a sequencer.

Amendments to the paragraph beginning at page 6, line 24:

Firstly, there is a problem that, because a microprocessor ~~mounted on~~ in a universal computer (hereafter referred to as universal Microprocessor), such as a personal computer, operates with an advanced function at a high speed and ~~can be~~ has become inexpensively available in recent years, the advantage of developing a microprocessor ~~mounted on~~ in a conventional programmable controller with an ASIC (Application Specific IC) is slowly ~~decreased~~ decreasing.

Amendments to the paragraph beginning at page 7, line 7:

Particularly, a universal microprocessor is provided with primary and secondary caches and constructed in accordance with ~~the~~ acceleration ~~mounting arts~~ of processing techniques such as pipeline processing, super-scaler processing and out-of-order processing and moreover, execution codes directly executable by the universal microprocessor are also generated by a compiler for converting the execution codes into execution codes forming an optimum code system by fully using the acceleration ~~mounting arts~~ techniques.

Amendments to the paragraph beginning at page 8, line 14:

Fourthly, though the "advanced-language sequence instruction program generator" disclosed in the above Japanese Patent Laid-Open No. HEI 7-295612 generates an object file by converting a source file described with the control program of an instruction list or the like into the source file of an advanced programming language such as C language, if only the source file of the advanced programming language is corrected. There is a problem that two source files showing the same sequential processing ~~are~~ do not ~~matched~~ match each other because the source file of an original control program is not changed due to the above correction.

Amendments to the paragraph beginning at page 8, line 25:

Particularly, in the case of a control-program-development supporting apparatus capable of developing a control program with an advanced programming language, to perform step execution by using a debugging tool, for example, step execution is possible for one line of the advanced programming language but it is impossible to perform step execution every line for a language configuration such as an instruction list before ~~converted~~ conversion into an advanced programming language.

Amendments to the paragraph beginning at page 9, line 23:

Fifthly, to realize the above-described under-RUN writing method, there is a problem that a memory having the same size as a memory in which a control program currently executed is read must be separately prepared. Because a memory is generally expensive among hardware parts, a maker or user must ~~pay much~~ make a large investment in order to realize the under-RUN writing method.

Amendments to the paragraph beginning at page 11, line 20:

~~Instruction~~An instruction interpreting unit is generally described with a structure of comparing input data with a pattern of the interpretation side and when the data and the pattern ~~are matched~~ match each other, executing a previously-stored operation (action) Therefore, pattern matching of the instruction interpreting unit requires a lot of time when the operation is executed in the case of the interpreter type and when the operation is compiled in the case of the compiling type. Thus, both types ~~are~~ have factors ~~for~~ impeding ~~the~~ productivity.

Amendments to the paragraph beginning at page 24, line 12:

Fig. 1 is a block diagram showing a schematic configuration of a programmable controller of a first embodiment;

Amendments to the paragraph beginning at page 24, line 18:

Fig. 3 is an illustration for explaining the generation of an execution code followed by division of ~~an~~ a control program in a control-program-development supporting apparatus of a second embodiment;

Amendments to the paragraph beginning at page 24, line 23:

Fig. 5 is a table showing the relation between the number of steps and a divided file in the control-program-development supporting apparatus of the second embodiment;

Amendments to the paragraph beginning at page 25, line 20:

Fig. 11 is an illustration for explaining the generation of an execution code followed by division of a control program and conversion of the program into an advanced programming language in a control-program-development supporting apparatus of a third embodiment;

Amendments to the paragraph beginning at page 26, line 17:

Fig. 15 is an illustration for explaining the generation of an execution code of a control-program-development supporting apparatus of a fourth embodiment;

Amendments to the paragraph beginning at page 26, line 21:

Fig. 16 is a table showing sample programs and their sizes, and processing times in a control-program-development supporting apparatus of a fifth embodiment;

Amendments to the paragraph beginning at page 26, line 25:

Fig. 17 is an illustration for explaining the development environment and execution environment of a control program in a control-program-development supporting apparatus of a sixth embodiment;

Amendments to the paragraph beginning at page 27, line 16:

Fig. 22A and Fig. 22B are illustrations for explaining the replacement of an execution code using a binary patch in a programmable controller of a seventh embodiment;

Amendments to the paragraph beginning at page 27, line 19:

Fig. 23 is an illustration for explaining the generation of an execution code in a control-program-development supporting apparatus of ~~eight~~ an eighth embodiment;

Amendments to the paragraph beginning at page 27, line 23:

Fig. 24 is an illustration showing a source file that is the content of a control program in the control-program-development supporting apparatus of the ~~eight~~ eighth embodiment;

Amendments to the paragraph beginning at page 28, line 2:

Fig. 24 is an illustration showing a source file that is the content of a control program in the control-program-development supporting apparatus of the ~~eight~~ eighth embodiment;

Amendments to the paragraph beginning at page 28, line 2:

Fig. 25 is an illustration showing a C-language expression obtained by converting a compressed file in the control-program-development supporting apparatus of the ~~eight~~ eighth embodiment;

Amendments to the paragraph beginning at page 28, line 6:

Fig. 26 is an illustration for explaining compiling in a control-program-development supporting apparatus of a ninth embodiment; and